A Probabilistic Shared-Control Framework for Mobile Robots

Soheil Gholami[1,2], Virginia Ruiz Garate[1], Elena De Momi[2], and Arash Ajoudani[1]

Abstract— **Full teleoperation of mobile robots during the execution of complex tasks not only demands high cognitive and physical effort but also generates less optimal trajectories compared to autonomous controllers. However, the use of the latter in cluttered and dynamically varying environments is still an open and challenging topic. This is due to several factors such as sensory measurement failures and rapid changes in task requirements. Shared-control approaches have been introduced to overcome these issues. However, these either present a strong decoupling that makes them still sensitive to unexpected events, or highly complex interfaces only accessible to expert users. In this work, we focus on the development of a novel and intuitive shared-control framework for target detection and control of mobile robots. The proposed framework merges the information coming from a teleoperation device with a stochastic evaluation of the desired goal to generate autonomous trajectories while keeping a human-in-control approach. This allows the operator to react in case of goal changes, sensor failures, or unexpected disturbances. The proposed approach is validated through several experiments both in simulation and in a real environment where the users try to reach a chosen goal in the presence of obstacles and unexpected disturbances. Operators receive both visual feedback of the environment and voice feedback of the goal estimation status while teleoperating a mobile robot through a control-pad. Results of the proposed method are compared to pure teleoperation proving a better time-efficiency and easiness-of-use of the presented approach.**

I. INTRODUCTION

Mobile manipulators, either wheeled such as MOCA [1], or legged such as ALMA [2], are becoming ubiquitous platforms in semi-structured (e.g., industrial [3]) or unstructured (e.g., urban [4]) environments. To operate such systems, several approaches have been proposed that range from teleoperation to autonomous control. In teleoperation, an operator, usually equipped with a motion capture system, generates movements for a follower robot to accomplish a desired task [1], [5]. Teleoperation systems have been a topic of research for a long time, with a particular focus on their stability and transparency [6]–[8]. In return, little to no attention has been given to the problems derived from their usability and intuitiveness of use [9]. Indeed, cognitive and physical fatigue are common issues [10] when users are faced with the operation of multi degrees-of-freedom (DoF) follower systems (see Fig. 1). Nonetheless, simpler systems with less DoF can also become bothersome during prolonged or repetitive tasks, e.g., for long-distance target reaching operations. On the other hand, fully autonomous systems provide a practical solution to these problems. Autonomous controllers are fast and effective and most suited for highly structured environments that are subject to few variations [11]. Still, they are highly vulnerable to unexpected changes in the environment and very dependent on the system's sensing capacities [12].

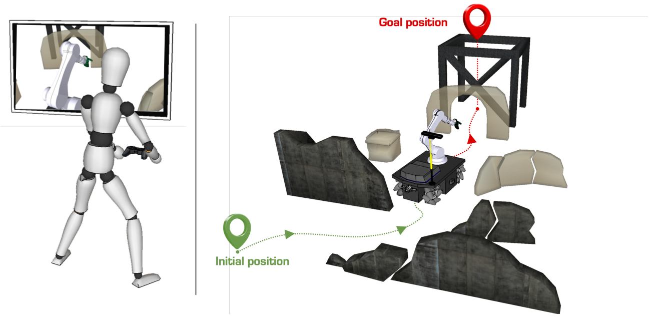


Fig. 1: Concept illustration for remote locomotion control of mobile robots in cluttered spaces. A pure teleoperation interface to navigate robots through long distances can be bothersome and frustrating.

To benefit from the advantages of both teleoperated and autonomous systems, several controllers have been proposed [13]–[17]. In [14], the authors developed a complex framework able to predict the human intention from visual data and to correct the robot motion based on preliminary learning algorithms that must be re-calculated to adapt to each user's movements. Moreover, the method must reinforce the user's teleoperation role as a way to avoid wrong inputs from their loss of motivation. Other works [15], [16] propose different levels of abstraction for the human user to command the desired objective of the robot, specifying particular goals and paths or high-level actions such as pick and place of an object. Nonetheless, these methods suffer from the lack of a human-in-control approach, meaning that the user is not able to interactively change the goal, and in the case of unexpected disturbances, the autonomous plan cannot be changed. A more complete method is presented in [17], where the user selects a command with an object-centered point-and-click approach from a visual Graphical User Interface (GUI), and the robot autonomously drives the action. During the autonomous execution, the user can react to unexpected errors by choosing new available actions on the GUI that affect robot navigation, camera position, etc. This type of approach requires very skilled users (in this case an astronaut) that are familiar with the GUI and the robot controls.

With the aim to simultaneously improve the performance and usability of teleoperation interfaces for mobile robots,

This work was supported by the European Research Council (ERC) starting grant Ergo-Lean, Grant Agreement No. 850932

[1] Human-Robot Interfaces and physical Interaction (HRI^2) Lab, Istituto Italiano di Tecnologia, Genoa, Italy, Email: Soheil.Gholami@iit.it

[2] NearLab, Dept. of Electronics, Information and Bioengineering, Politecnico di Milano, Milan, Italy

in this manuscript, we propose an intuitive shared-control framework based on a probabilistic target estimation algorithm. This allows the operator to progressively drive the robot towards the desired goal while interfacing through a control-pad device and receiving feedback from a GUI. Once the goal is detected, the robot drives autonomously towards it. This goal can be changed dynamically by the user while the controller adapts seamlessly and autonomously between the different needs, always maintaining a human-in-control structure. This means that at any point, the operator can take full control of the platform, changing the goal or modifying the generated trajectory because of unexpected disturbances.

The controller is also capable of autonomous obstacle avoidance through real-time laser data. In the case where laser information cannot detect fast disturbances (e.g., sudden human movements, or in presence of transparent objects along its path), the user can easily and immediately take control, avoid the obstacles and switch back to the autonomous mode. The framework is validated through virtual and real-world experiments in which several users drive a mobile robot towards a desired goal in the presence of obstacles and unexpected disturbances.

II. METHODOLOGY

The overall structure of the proposed shared-control framework is shown in Fig. 2. The user starts the navigation in the *teleoperation* mode with full control over the robot's motion. Meanwhile, the *target estimation* block looks for available targets (if any), and evaluates the user's desired goal through laser and odometry sensors. Once the target pose $\mathbf{p}_g = [x_g, y_g, \theta_g]^T$ is estimated, the user can release the control-pad device. This activates the *navigation stack* where the robot moves autonomously towards the target. If an obstacle appears in the robot's path, the obstacle-avoidance algorithm embedded in the navigation stack tries to recognize it and re-plan the trajectory towards the goal. If the algorithm fails to detect the obstacle, our framework allows the user to take back the full control over the robot motion and navigate around the obstacle safely in the teleoperation mode. To retake the control of the platform, the user simply needs to touch the motion buttons of the control-pad. In this case, the robot's autonomous movement stops and the user controls it directly to a safe location. The *autonomous* mode then navigates the mobile platform towards a newly identified goal by checking the goal's likelihood (if no target is found, the user should further explore the environment with the teleoperation mode). A detailed description of each element of the control framework is given hereafter.

A. State monitor

The *state monitor* block reads the status of the control-pad, and transforms it into meaningful information for the framework. The percentage of the desired linear v_p and angular ω_p velocities are updated by the left and right dual-shockers of the pad, respectively. While the user controls the platform's motion, the *mode* flag is triggered indicating that the platform is in teleoperation mode.

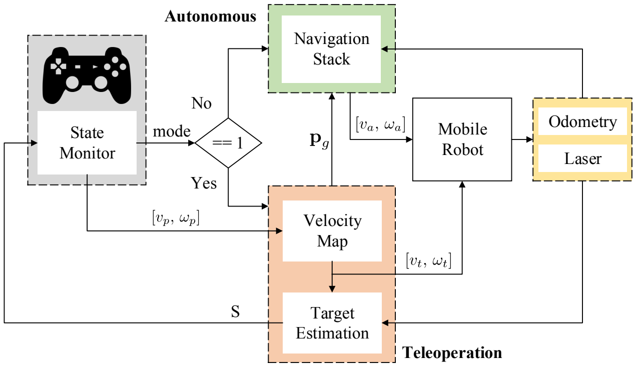


Fig. 2: The block diagram of the proposed shared-control framework. The user actuates the robotic platform by means of a control-pad. During teleoperation ($mode = 1$), the developed algorithm estimates his/her intention simultaneously. Once a target has been recognized ($S = 1$), the user can release the pad and the robot reaches the goal autonomously ($mode = 0$).

If the user releases the control-pad, the mode flag resets and based on the *estimation status* S, the state monitor decides about the next phase. In this situation, if the target goal is estimated ($S = 1$), the mode flag is reset and the autonomous part is activated. This block autonomously controls the motion of robot towards the new target pose (received from the *target estimation* block).

B. Teleoperation mode

During the teleoperation mode (Fig. 2), two main operations are performed: (i) the inputs from the user are sent to the mobile robot through the *velocity map* block, and (ii) the most likely target is constantly being estimated based on the input commands and the processed laser sensor data.

The velocity map block converts the received data to the robot's twist command ($[v_t, \omega_t]$) based on the maximum values allowed for the mobile platform $v_t = v_p\, v_{max}$ and $\omega_t = \omega_p\, \omega_{max}$.

The target estimation block is in charge of evaluating the user intention based on the laser sensor data and the commands being sent by the user through the pad. As illustrated in Fig. 3, while the robot, with frame $\{R\}$, moves in an area with respect to (w.r.t.) the fixed frame (odometry frame) $\{O\}$, its laser sensor scans a subsection of that area within a specific range ($r_{min} < r < r_{max}$) and angle ($\phi_{min} < \phi < \phi_{max}$). As a result, a 2D point cloud of the environment is generated, whose points are defined in the polar coordinates of the laser frame $\{L\}$ as ${}^{\mathrm{L}}\mathbf{p}_j = [r_j, \psi_j]^T$. At each scanning step, if there is no point in one of the laser measurements, $r_j = +\infty$. This is used to classify the points into different 2D clusters ($\boldsymbol{\gamma}_i$). Still, in some cases two different clusters might be classified as one because of their proximity (in polar angle). To handle this issue, a threshold r_{tsh} is defined between two consecutive points r_j and r_{j+1} of the scan stream. If $|r_j - r_{j+1}| > r_{tsh}$, the new point is considered as part of a new cluster.

The centroid of the i-th cluster $\boldsymbol{\gamma}_i$, ${}^{\mathrm{L}}\mathbf{c}_{\gamma_i} = [x_{\gamma_i}, y_{\gamma_i}]^T$, is obtained by averaging the cluster's points given by the laser ($\mathbf{x}_{\gamma_j}$, $\mathbf{y}_{\gamma_j}$), where $j = 1 \cdots \rho_{\gamma_i}$, being ρ_{γ_i} the number

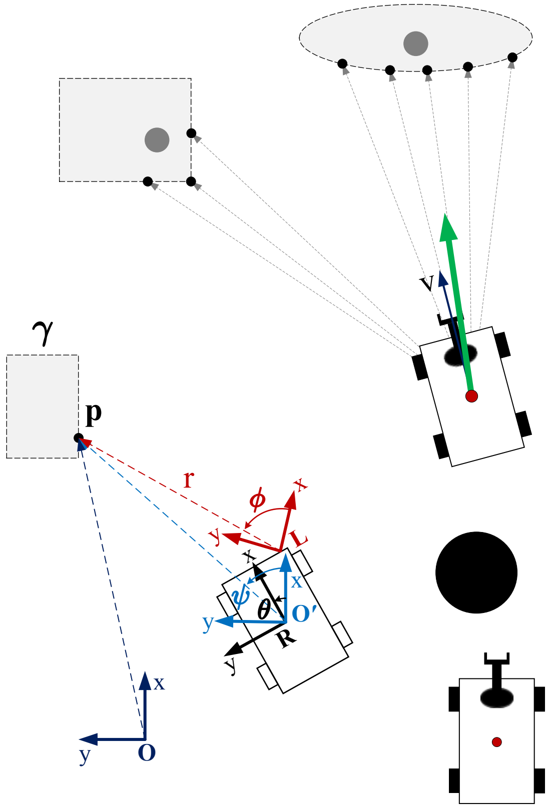


Fig. 3: Diagram of the different Cartesian frames during navigation. Some graphical elements have been distributed among the different robot poses for clarity. New clusters are recognized by the laser sensor in its frame $\{L\}$ (in red) while the robot moves and transformed into the $\{O'\}$ frame (in light blue). The pose of these clusters in $\{O'\}$ is analyzed w.r.t. that of the robot $\{R\}$ (in black) to estimate the most likely target (green arrow in the upper robot drawing). The pose of the clusters are stored in the fixed frame $\{O\}$ (in dark blue) to create a fix database that enables the comparison of already recorded and new clusters.

of points in the cluster i. These are transformed to frame $\{O\}$ using the transformation matrix between both frames: ${}^{\mathrm{O}}\mathbf{c}_{\gamma_i} = {}^{\mathrm{O}}\mathbf{T}_{\mathrm{L}}\,{}^{\mathrm{L}}\mathbf{c}_{\gamma_i}$, being ${}^{\mathrm{O}}\mathbf{T}_{\mathrm{L}} = {}^{\mathrm{O}}\mathbf{T}_{\mathrm{R}}\,{}^{\mathrm{R}}\mathbf{T}_{\mathrm{L}}$. ${}^{\mathrm{R}}\mathbf{T}_{\mathrm{L}}$ is a constant matrix given by the robot dimensions and the laser placement and ${}^{\mathrm{O}}\mathbf{T}_{\mathrm{R}}$ is given by the odometry measurements of the robot. This centroid is recognized as a new one if its Euclidean distance to all of the previously stored centroids is greater than a specific threshold: $||{}^{\mathrm{O}}\mathbf{c}_{\gamma_i} - {}^{\mathrm{O}}\mathbf{c}_{\gamma_{1,\cdots,n}}|| > \gamma_{tsh}$, being n the number of clusters. As a result, the centroid and the minimum and maximum values of the xy coordinates of the cluster's points (x_i, y_i), are added to a framework's look-up table $\mathbf{\Gamma}$:

$$\mathbf{\Gamma} = \begin{bmatrix} x_{\gamma_1} & x_{\gamma_2} & \cdots & x_{\gamma_n} \\ y_{\gamma_1} & y_{\gamma_2} & \cdots & y_{\gamma_n} \\ \mathbf{x}_{1,min} & \mathbf{x}_{2,min} & \cdots & \mathbf{x}_{n,min} \\ \mathbf{x}_{1,max} & \mathbf{x}_{2,max} & \cdots & \mathbf{x}_{n,max} \\ \mathbf{y}_{1,min} & \mathbf{y}_{2,min} & \cdots & \mathbf{y}_{n,min} \\ \mathbf{y}_{1,max} & \mathbf{y}_{2,max} & \cdots & \mathbf{y}_{n,max} \end{bmatrix}. \qquad (1)$$

Once the clusters' look-up-table $\mathbf{\Gamma}$ has been built, the target estimation problem is formulated from a probabilistic approach. To be able to compare the robot orientation to the clusters, a frame $\{O'\}$ is considered, that is the result of translating $\{O\}$ to the origin of $\{R\}$ (see Fig. 3). At each instance k, the robot's heading angle $\theta(k)$ is given by the robot odometry sensors w.r.t. the z axis of $\{O'\}$. Besides, the j-th detected point of the cluster $\boldsymbol{\gamma}_i$, can be expressed in $\{O'\}$ using the corresponding transformation matrices: $[{}^{\mathrm{O'}}\mathbf{p}_{\gamma_{i,j}}, 1]^T = {}^{\mathrm{O'}}\mathbf{T}_{\mathrm{L}}\,[{}^{\mathrm{L}}\mathbf{p}_{\gamma_{i,j}}, 1]^T$, with ${}^{\mathrm{O'}}\mathbf{T}_{\mathrm{L}} = {}^{\mathrm{O'}}\mathbf{T}_{\mathrm{R}}\,{}^{\mathrm{R}}\mathbf{T}_{\mathrm{L}}$. ${}^{\mathrm{O'}}\mathbf{T}_{\mathrm{R}}$ is constructed from the odometry measurement of the robot corresponding to the rotation angle, as $\{O'\}$ is translated to the robots frame (Fig. 3). Moreover, the direction of each vector ${}^{\mathrm{O'}}\mathbf{p}_{\gamma_{ij}}$ w.r.t. the z axis of $\{O'\}$, $\psi_{\gamma_{ij}}(k)$, is obtained by using the trigonometric functions:

$$\psi_{\gamma_{ij}}(k) = tan^{-1}\left({}^{\mathrm{O'}}\mathbf{p}_{\gamma_{i,j}\,y},\ {}^{\mathrm{O'}}\mathbf{p}_{\gamma_{i,j}\,x}\right). \qquad (2)$$

At each instant k, direction angles of the i-th cluster are stored in the corresponding vector $\boldsymbol{\psi}_{\gamma_i}(k)$. To calculate the cluster's mean $\mu_{\gamma_i}(k)$ and standard deviation $\sigma_{\gamma_i}(k)$, the following equations are used:

$$\begin{aligned} \mu_{\gamma_i}(k) &= \frac{1}{\rho_{\gamma_i}} \sum_{j=1}^{\rho_{\gamma_i}} \psi_{\gamma_{i,j}}(k), \\ \sigma_{\gamma_i}(k) &= \left(\frac{1}{\rho_{\gamma_i}} \sum_{j=1}^{\rho_{\gamma_i}} \left[\psi_{\gamma_{i,j}}(k) - \mu_{\gamma_i}(k)\right]\right)^{\frac{1}{2}}. \end{aligned} \qquad (3)$$

To account for the possible changes along time in the points detected for the cluster $\boldsymbol{\gamma}_i$, the average value of the standard deviation over the last N samples is computed:

$$\bar{\sigma}_{\gamma_i}(k) = \frac{1}{N} \sum_{i=k-N+1}^{k} \sigma_{\gamma_i}. \qquad (4)$$

The likelihood that the user target g belongs to a particular cluster $\boldsymbol{\gamma}_i$ is given by [18]:

$$L(g|\boldsymbol{\gamma}_i)(k) = \frac{1}{\sqrt{2\,\pi}\,\bar{\sigma}_{\gamma_i}(k)}\, e^{\frac{-1}{2\,\bar{\sigma}^2_{\gamma_i}(k)}\,\delta^2_{\gamma_i}(k)}. \qquad (5)$$

$\delta_{\gamma_i}(k)$ is the average error between the robot's heading angle θ and the mean of the cluster points directions μ_{γ_i} over the last N samples:

$$\delta_{\gamma_i}(k) = \left(\frac{1}{N} \sum_{i=k-N+1}^{k} \left[\theta(k) - \mu_{\gamma_i}(k)\right]^2\right)^{\frac{1}{2}} \qquad (6)$$

If the maximum likelihood value among all clusters $L(g|\boldsymbol{\gamma}_i)(k)$ is greater that a threshold L_{tsh}, the cluster $\boldsymbol{\gamma}_i$ is selected as the desired goal of the user. This target is estimated only based on the robot's heading angle, which is governed by the angular twist velocity. To determine the user's intention to actually go towards the target, the average linear velocity over the last N samples is calculated:

$$\bar{v}(k) = \frac{1}{N} \sum_{i=k-N+1}^{k} v(k), \qquad (7)$$

If $\bar{v}(k)$ is greater than a threshold v_{tsh}, the estimated target is activated and the user can release the control-pad device.

C. Autonomous mode

Once the *autonomous* block (Fig. 2) is activated, the robot navigates autonomously towards the estimated target pose $\mathbf{p}_g = [x_g,\, y_g,\, \theta_g]^T$. To do so, we use the algorithm presented in [19], [20] which employs the timed-elastic-bands (TEB) concept to have an efficient online path and motion planning. In their optimization problem formulation, transition time, path length, and control efforts are considered as the minimizing objectives, and the clearance from the obstacles, which are set of points, circles, lines, and polygons, is the constraint of the optimization problem. To create the set of continuous points out of the available obstacles, the algorithm uses the laser scan data to calculate the Euclidean distance between the robot and each obstacle, and in turn, the controller generates the deviating control efforts to bypass the obstacle in an optimized manner.

Therefore, the *navigation stack* takes in the information from the robot's odometry sensors, laser sensor (to build the environment map and avoid the obstacles), and the goal $\mathbf{p}_g$. As an output, it gives the safe and optimized twist commands which are then sent to the mobile platform $[v_a, \omega_a]$. During the navigation task, the obstacle detection algorithm [20] enables the robot to avoid static and dynamic obstacles. If an obstacle is not detected, the human-in-control approach of our framework allows the user to come back to the teleoperation mode controlling the motion of the robot.

III. EXPERIMENTS

Several experiments are performed to validate the performance and intuitiveness-of-use of the proposed shared-control framework focusing on the remote navigation control of the mobile robots. In the first step, a complex virtual environment envisaging multiple goals, changes between them, and unexpected disturbances are tested. In this scenario, we compare our method with a full teleoperation one to examine the advantages of the shared-control approach. Second, we develop a set of real-world experiments to validate the introduced framework in the presence of disturbances and sensor failures. In all the experiments, the goals are defined as the position of the target face nearer to the user with 0 deg orientation.

For the real experiments, we use the mobile manipulator MOCA [1], which is an integration of a SUMMIT-XL STEEL mobile base, a Franka Emika manipulator, and the under-actuated Pisa/IIT soft hand as end-effector [21]. On the other hand, a virtual model of SUMMIT-XL STEEL mobile base[1] is considered in the simulation experiments. To control the robot in the teleoperation mode a Sony PlayStation4 dual-shock control-pad device is used in both virtual and real experiments. Moreover, the controller (Fig. 2) is implemented in C++ using the ROS framework.

In the following, both experimental sets are described and then the results are given. A discussion is added to evaluate the performance of the experiments.

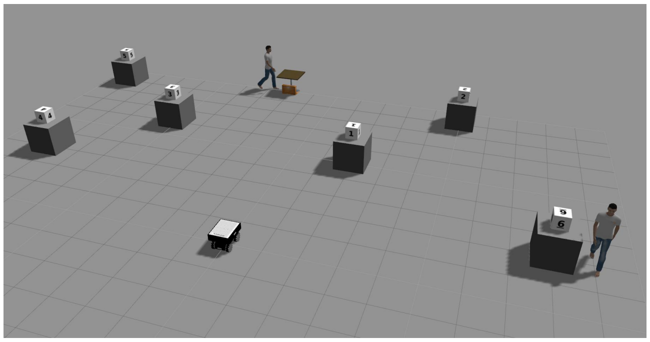


Fig. 4: The simulation environment in Gazebo. The user controls the virtual model of the SUMMIT-XL STEEL mobile robot towards his/her desired goals. Six target goals (boxes) are considered in this environment: $b_1 = [5, -2]$ m, $b_2 = [9, -5]$ m, $b_3 = [6, 5]$ m, $b_4 = [3, 8]$ m, $b_5 = [9, 9]$ m, and $b_6 = [2, -8]$ m.

.

A. Multi-target experiments in the virtual environment

The virtual environment is developed within the Gazebo software, where a virtual model of the SUMMIT-XL STEEL platform is utilized to simulate the mobile robot. Fig. 4 displays a screenshot of such environment. The area considered is 10×10 m and six different boxes of size $1 \times 1 \times 1$ m are placed as possible targets. This simulates a typical industrial set-up, where the possible target locations are quite structured. Nonetheless, obstacles and unexpected disturbances are likely to happen, specially if humans are present in the working environment.

The different targets are divided into two main groups (left and right targets in Fig. 4) to test two different kind of disturbances: on the left side, fixed obstacles, and on the right, dynamic (unexpected) obstacles. In the first case, obstacle avoidance is performed in an autonomous way. We forecast that the proposed multi-target estimation method will have a higher performance than the full teleoperation one. In the second case, we aim to additionally prove the usefulness of the shared-control with the human-in-control approach, as the autonomous control is not able to react fast enough to the disturbance and teleoperation (human decision making) is needed.

12 subjects participated in the virtual experiments and performed 4 different trials: reaching 3 left-side targets with full teleoperation (LT) and shared-control (LS) with fixed obstacles, and reaching 3 right-side targets with full teleoperation (RT) and shared-control (RS) with unexpected obstacles. Each of the 4 trials is executed once by each subject. To render the results as less biased as possible, the order between the full teleoperation and shared-control trials is randomized, also starting arbitrarily from the left or right targets. Nonetheless, we decided not to alternate between teleoperation and shared-control methods for the same subject. This decision is made to reduce the adaptation needed between both methods and to simplify the subjective questionnaires (see Section III-C). For each trial, the operator is commanded to reach a particular sequence of goals whose order is determined randomly. The same order of targets is used for the teleoperated and the shared-control cases to allow a better comparison of execution times and errors. The

[1] www.github.com/RobotnikAutomation/summit_xl_sim

TABLE I: Age, sex, trials and target order b_i for the left L and right R targets for each subject S_i.

Subject	Age	Sex	Trials	L targets	R targets
S_1	36	F	LT-RT–LS-RS	b_3-b_4-b_5	b_1-b_2-b_6
S_2	37	M	RT-LT–LS-RS	b_4-b_3-b_5	b_2-b_6-b_1
S_3	26	M	LS-RS–RT-LT	b_4-b_3-b_5	b_6-b_2-b_1
S_4	56	M	RS-LS–RT-LT	b_5-b_4-b_3	b_1-b_2-b_6
S_5	27	M	LT-RT–LS-RS	b_4-b_3-b_5	b_1-b_6-b_2
S_6	25	F	RT-LT–LS-RS	b_4-b_5-b_3	b_2-b_6-b_1
S_7	25	M	LS-RS–RT-LT	b_3-b_4-b_5	b_2-b_6-b_1
S_8	26	F	RS-LS–RT-LT	b_5-b_3-b_4	b_6-b_1-b_2
S_9	31	F	LT-RT–LS-RS	b_5-b_4-b_3	b_1-b_2-b_6
S_{10}	30	M	RS-LS–LT-RT	b_4-b_3-b_5	b_1-b_2-b_6
S_{11}	30	M	LS-RS–LT-RT	b_5-b_4-b_3	b_2-b_1-b_6
S_{12}	34	M	RT-LT–RS-LS	b_4-b_3-b_5	b_1-b_6-b_2

obstacles are represented by a sphere of 1 m diameter placed randomly between targets. The same position for the obstacle is kept for the teleoperated and the shared-control trials. For the right-side targets, as obstacles must appear dynamically, the position in the first mode is stored to later release it again in the same position. The characteristics of the subjects performing the experiments and the randomized variables are presented in Table I. Subjects are allowed to get used to the environment and the controllers in the preferred mode for 3 minutes before the start of the experiments. 7 of the 12 subjects declared themselves as "experts" whereas 5 subjects had not previous or very little experience with control-pad devices.

For these trials, the control parameters are selected as follows: $v_{max} = 0.35\,m/s$, $\omega_{max} = 0.30\,rad/s$, $[r_{min}, r_{max}] = [0.5\,m, 8.0\,m]$, $[\phi_{min}, \phi_{max}] = [-\pi, \pi]$, $\mathbf{v}_{tsh} = 0.2\,m/s$, $L_{tsh} = 1.5$, $r_{tsh} = 1.0$, and $N = 25$. The maximum velocities are set to allow the subjects to react to possible obstacles while being fast enough not to render the experiments tedious. The threshold values are chosen empirically based on the previous experiments and analysis of the processed data.

B. Real-world implementation

Two sets of experiments are carried out in a laboratory setting. These aim to prove the validity of the approach in a real-world scenario where inaccuracies and sensors' faults are more frequent. The experimental set-up is shown in Fig. 5. The MOCA mobile manipulator starts from the origin within an area of 10 m length and 6 m width. Several objects are placed in the environment which can be considered as the target goals. When the experiment starts, the laser sensor scans the environment constantly. The initial laser scan of the environment is displayed in Fig. 6 which keeps generating a 2D point cloud of the environment until the task is done. The locations of the targets are estimated by means of the constant update of these detected points (Section II).

For these set of experiments, the user is instructed to go first to the nearer right target and then to the left one (see Fig. 5). by using the control-pad device while the shared-control framework is activated. Once the target is estimated, the user is informed to release the control-pad device after which the robot goes to the target autonomously. For the rest of the navigation (from right target to left one), 4 different scenarios are considered. In the first two scenarios (Set 1), a laser-detectable object is considered which is suddenly pushed into the robot's path. This is done by placing a box not high enough for the laser to detect it. In the other two scenarios, a non-detectable object is employed (Set 2) which is in a fixed position. Each set starts with the human-in-control obstacle avoidance in which the user reacts to the sudden obstacle (Set 1) or the detection failure (Set 2). These trials are later repeated allowing the controller to perform the autonomous obstacle-avoidance.

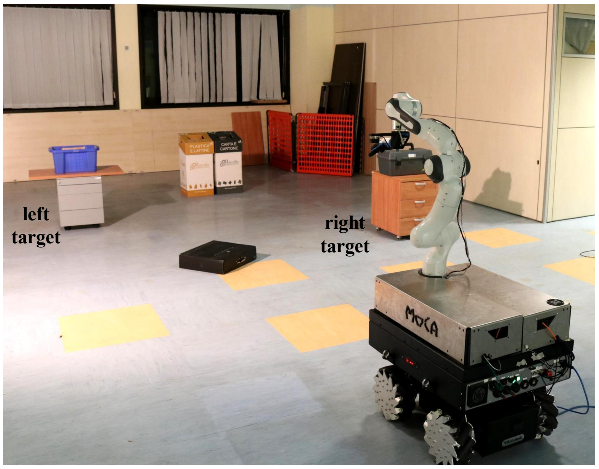


Fig. 5: The laboratory experimental setting considered for the real-world implementation. The MOCA mobile manipulator starts from the displayed position in all trials. Several objects are placed in the environment which can be considered as the target goals.

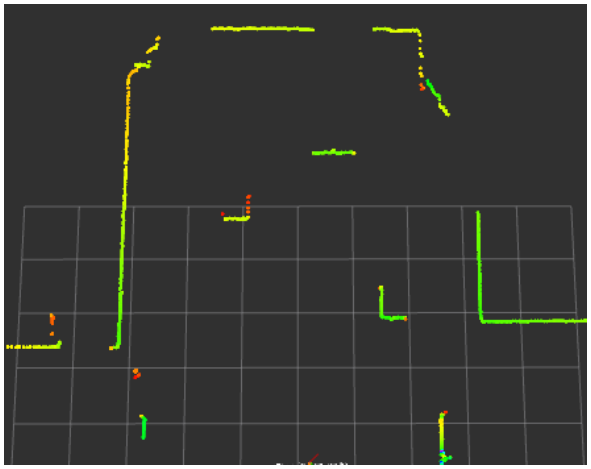


Fig. 6: The initial laser sensor scan result creating the 2D point cloud of the environment. The locations of the targets are estimated by means of the constant update of these detected points.

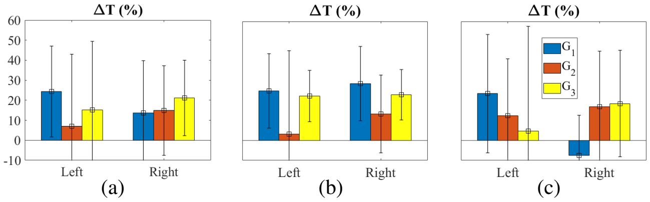
For these experiments, the control parameters are set as: $v_{max} = 0.25\,m/s$, $\omega_{max} = 0.20\,rad/s$, $[r_{min}, r_{max}] = [0.5\,m, 8.0\,m]$, $[\phi_{min}, \phi_{max}] = [-\pi, \pi]$, $\mathbf{v}_{tsh} = 0.15\,m/s$, $L_{tsh} = 1.5$, $r_{tsh} = 1.0$, and $N = 25$. Here, for safety reasons, lower values for velocities are chosen, and accordingly, the velocity threshold v_{tsh} is decreased to 0.15.

C. Experimental results

In this section, the results of the virtual and real-world experiments are presented.

TABLE II: the mean (μ) and standard deviation (σ) of the task execution time, teleoperation mode percentage, and the target pose error in the simulation experiments.

			LT	LS	RT	LS
Task execution time [s]		μ	42.12	37.45	41.09	34.80
		σ	15.64	26.11	11.17	15.27
Time in teleoperation mode (%)		μ	100	43.52	100	41.41
		σ	0	20.05	0	19.94
Pose error	x [m]	μ	0.046	0.14	0.075	0.12
		σ	0.15	0.045	0.17	0.019
	y [m]	μ	0.073	0.13	0.056	0.068
		σ	0.051	0.16	0.039	0.041
	θ [deg]	μ	2.18	3.062	1.87	3.31
		σ	2.96	2.60	1.92	2.27



(a) (b) (c)

Fig. 7: Increment in the execution time when comparing full teleoperation and shared-control approaches. $\Delta T(\%)$ represents the execution time difference between two control modes, being the shared-control $\Delta T(\%)$ faster than the pure teleoperation mode. The Left and Right scenarios are separated. As the goal numbers are randomized during each mode, each G_i indicate the goal's reaching order, e.g. G_1 is the first goal reached which can be any of the b_i. (a) Overall results. (b) Results for "experienced" users. (c) Results for "naive" users.

First, a quantitative analysis of the virtual experiments is performed. Three variables are evaluated: pose error between the desired final pose ($\mathbf{p}_g$) and the reached one by the platform, time needed to execute the task (task execution time), and percentage of time that users where employing the teleoperation (for the shared-control trials). These variables are compared between the pure teleoperated mode and the proposed shared-control framework. The mean (μ) and standard deviation (σ) of these variables are displayed in Table II. Additionally, Fig. 7 shows the execution time comparison in the order of the goals to be reached (which changes randomly for each subject). This figure also shows the differences between the "expert" and "naive" users.

Fig. 8 shows an example case of the evolution of the control signals over time for one possible virtual task (target sequence: $b_4 - b_3 - b_5$). As it can be observed, the user starts the task in the teleoperation mode at $k \approx 200$ ($mode = 1$). Consequently, the likelihood value of each target is constantly being updated while navigating the robot in the environment ($L(g|\gamma_i)$, $i = 3, 4, 5$). When the user tries to drive the robot towards the first goal (b_4), the value of $L(g|\gamma_3)$ (black line) increases due to the robot's heading angle getting closer to b_3. However, the robot's linear velocity (v) is not greater than the specified threshold v_{tsh}, so the estimation is not activated for b_3 and the algorithm continues to estimate the goals. At $k \approx 650$, the $L(g|\gamma_4)$ (red line) and v are greater than L_{tsh} and v_{tsh}, respectively. Thus, the estimation is activated and the user, who is informed by the voice feedback, releases the control-pad device ($mode = 0$). The robot then goes autonomously towards the target. When the robot reaches its goal ($k \approx 1700$), the user moves the robot backwards between $1800 < k < 1900$ (negative values in v) to select another goal, and repeats this cycle to end the task. It should be noted that in each phase of the task, an obstacle is placed on the way of robot's path. This leads to the current active goal's likelihood becoming approximately 0. (e.g., $L(g|\gamma_4) \approx 0$ in $950 < k < 1400$)

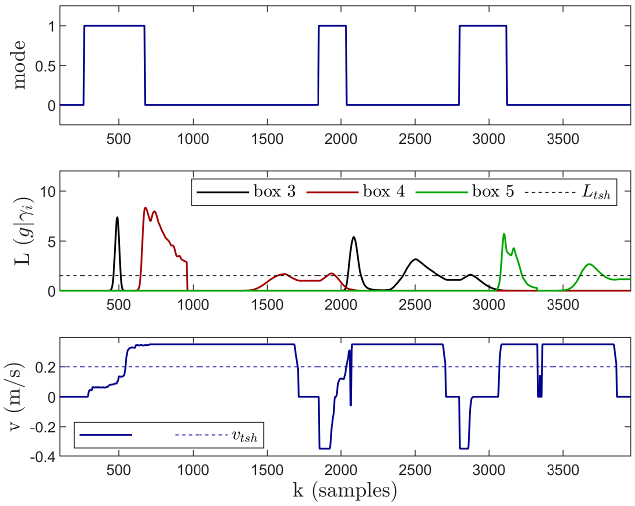


Fig. 8: The evolution of the control variables in time (the logger sample-rate is 50 Hz). There are two threshold values, one for likelihood L_{tsh} and the other for the platform's linear velocity v_{tsh}. If these thresholds are satisfied at a specific moment, the user is informed by voice feedback and he/she can release the control-pad device. The mode variable is 0 when the user releases the device.

Finally, a subjective questionnaire is carried out to evaluate the perceived effort, usefulness, and demand of both methods. This questionnaire includes the Official NASA Task Load Index (TLX) [22] survey, plus an additional one-question test. The latter was chosen to be as simple as possible, giving an overview of the methods. The question is: *Overall, this task was?*, and users can rate it from *1- Very difficult* to *7- Very easy*. The whole survey is filled by the subjects twice: once after the full teleoperation trials and once after the shared-control ones. The results of NASA TLX are presented in Fig. 9. In addition, the results of the one-question test, which is normalized to 100%, give a 67.86% of easiness to the full teleoperation, and 89.28% to the shared-control mode.

Regarding the real experiments, Fig. 10 shows the position of the robot over time while executing the 4 described scenarios. For both kind of obstacles, when the user takes control the robot can reach the goal. When the autonomous controller is used, even if the obstacle is detected and the avoidance algorithm re-plans the trajectory, the platform still hits the obstacle from one corner (Fig. 10b). If the obstacle is not detected (Fig. 10d), the algorithm does not re-plan the trajectory. In this case, since the obstacle is a box, it is pushed along the path until the emergency button is pressed.

D. Discussion

As it may be seen in Table II, the overall time to execute the trial is lower for the shared-control than for the full

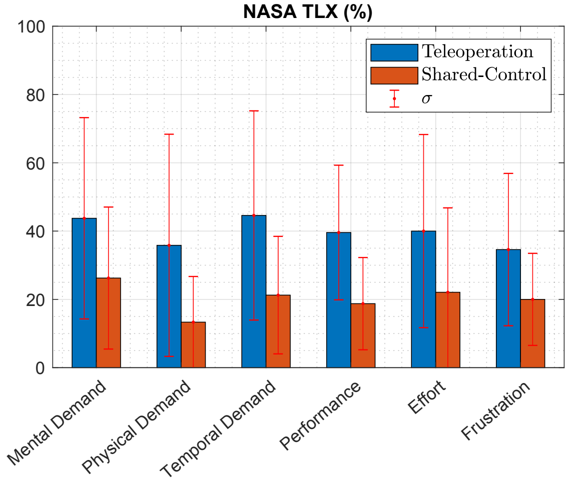


Fig. 9: The result of NASA TLX test comparing full teleoperation and shared-control methods in a navigation task.

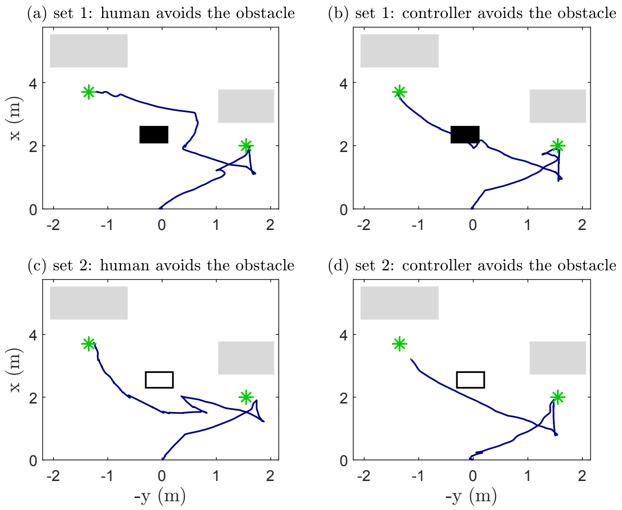


Fig. 10: Traces of the MOCA's odometry while accomplishing the tasks in the real world experiment. Gray rectangles show the target and the green ⋆ shows the safe position of the robot before the estimated pose. We used two sets of experiments; in the first one, a detectable obstacle (black rectangle) is thrown to the path of the robot while in the second one, a non-detectable obstacle (white rectangle) is set in the area ahead of the robot. In both sets, the user was first asked to avoid the obstacle himself/herself (left figures). After that, we repeated the similar scenarios while this time, the obstacle avoidance algorithm takes care of avoiding the obstacle.

teleoperation scenarios, which is related to the use of the autonomous (and therefore faster) mode. For the left trials, shared-control is 15.51% faster than teleoperation, while for the right ones is 16.56% faster. However, as it can be seen in Fig. 7, for the case of unexpected obstacles (right trials), when the users have to teleoperate around it, they seem to learn to take advantage of the shared-control over time, being the difference bigger in the last goal. This learning is evident in the "naive" users, but for the "expert" users the trend is the opposite. This means that while the "naive" users clearly learn to take advantage of the autonomous mode (from a negative difference to a bigger improvement) over time, the "expert" users produce more random patterns. With the fixed obstacles (left trials), while all the users are able to execute the task faster with the shared-control, "naive" ones tend to rely less on the autonomous mode over time, which could be related with them getting used to the teleoperation device. However, the final pose accuracy (Table II) is lower in shared-control. This is due to the fact that the desired goal is estimated from the detected points of the cluster, and therefore, subject to errors, while for the teleoperation mode the subjects used the visual feedback that provided them with a more accurate position of the target. Nonetheless, the errors in position are always kept below than 10 cm and 3.5 deg.

In addition, the amount of time in which the users drive the platform through the control-pad during the proposed method (LS, RS - Table II) accounts for less than 50% of the total execution time. This means that even if teleoperation mode is needed to detect the target, the user still favored the autonomous mode when possible.

Results from the subjective questionnaires show an outperforming shared-control. As expected from the quantitative results, the perceived temporal demand is higher in the full teleoperated than in shared-control trials (see Fig. 9). Moreover, overall, the subjects perceive less physical and psychological demand with our proposed method. Users also report better performance during the shared-control trials (note that a higher index means more perceived failure), while feeling less effort and frustration. The one-question test confirmed these results as users give a higher mark (meaning easier to execute) to our method than to the full teleoperated one. It must be pointed out that these experiments took less than 30 mins for each subject, so bigger differences could be expected for longer times, i.e., in normal working conditions. The high standard deviation present all along the results is due to the high randomization of the experiments and the variability of the users' profiles.

Finally, real-world experiments validate the functioning of the proposed framework. Furthermore, they show the need for the human-in-control approach during the shared-control, as when obstacles are unexpected or the sensors fail to detect them, the robot collides with the obstacles if the human does not take charge of the motion of the platform.

IV. CONCLUSION

In this work, we introduced a shared-control framework for the locomotion control of mobile manipulators in the presence of unexpected disturbances. The developed framework proved to successfully identify the target goals by using the robot's odometry and laser data. Moreover, the users could take over the control of the robot at any time, e.g., when an obstacle was observed through the monitor which may have been neglected by the autonomous obstacle detection system due to the sensor failures or the rapid change of environment's features. After releasing the control-pad device, the controller smoothly switched back to the autonomous mode and navigated the mobile platform towards the previously identified goal. Similarly, it was proven that the user could dynamically change the goal by executing a continuous path of several targets.

Overall, the experiments revealed the time-efficiency and ease-of-use of the proposed interface, with a low perceived cognitive effort and a high-performance estimation.

The proposed framework can be easily extended to cope with manipulation tasks (employing 3D point cloud data), as the targets are continuously updated and can be used to command the loco-manipulation unit from the end-effector instead of the base. Therefore, future works will concentrate on the manipulation aspects towards the development of whole-body shared-control of mobile manipulators. Furthermore, enhanced activation techniques such as image augmentation will be studied to highlight the current estimated target in highly cluttered environments and to notify the user about detected obstacles.

REFERENCES

[1] Y. Wu, P. Balatti, M. Lorenzini, F. Zhao, W. Kim, and A. Ajoudani, “A teleoperation interface for loco-manipulation control of mobile collaborative robotic assistant,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3593–3600, Oct 2019.

[2] C. D. Bellicoso, K. Krämer, M. Stäuble, D. Sako, F. Jenelten, M. Bjelonic, and M. Hutter, “Alma - articulated locomotion and manipulation for a torque-controllable robot,” in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 8477–8483.

[3] ©KUKA AG 2020, “KMR iiwa,” https://www.kuka.com/en-gb/products/mobility/mobile-robots/kmr-iiwa, 2020, [Online; accessed 17-February-2020].

[4] N. G. Tsagarakis, D. G. Caldwell, F. Negrello, W. Choi, L. Baccelliere, V. Loc, J. Noorden, L. Muratore, A. Margan, A. Cardellino, L. Natale, E. Mingo Hoffman, H. Dallali, N. Kashiri, J. Malzahn, J. Lee, P. Kryczka, D. Kanoulas, M. Garabini, M. Catalano, M. Ferrati, V. Varricchio, L. Pallottino, C. Pavan, A. Bicchi, A. Settimi, A. Rocchi, and A. Ajoudani, “Walk-man: A high-performance humanoid platform for realistic environments,” *Journal of Field Robotics*, vol. 34, no. 7, pp. 1225–1259, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21702

[5] K. Darvish, Y. Tirupachuri, G. Romualdi, L. Rapetti, D. Ferigo, F. J. A. Chavez, and D. Pucci, “Whole-body geometric retargeting for humanoid robots,” *arXiv preprint arXiv:1909.10080*, 2019.

[6] T. B. Sheridan, “Teleoperation, telerobotics and telepresence: A progress report,” *Control Engineering Practice*, vol. 3, no. 2, pp. 205–214, 1995.

[7] M. Selvaggio, P. R. Giordano, F. Ficuciellol, and B. Siciliano, “Passive task-prioritized shared-control teleoperation with haptic guidance,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 430–436.

[8] D. Sun, F. Naghdy, and H. Du, “A novel approach for stability and transparency control of nonlinear bilateral teleoperation system with time delays,” *Control Engineering Practice*, vol. 47, pp. 15 – 27, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066115300447

[9] M. Laghi, M. Maimeri, M. Marchand, C. Leparoux, M. Catalano, A. Ajoudani, and A. Bicchi, “Shared-autonomy control for intuitive bimanual tele-manipulation,” in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2018, pp. 1–9.

[10] J. V. Draper and L. M. Blair, “Workload, flow, and telepresence during teleoperation,” *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1030–1035 vol.2, 1996.

[11] J. Jiao, S. Ye, Z. Cao, N. Gu, X. Liu, and M. Tan, “Embedded vision-based autonomous move-to-grasp approach for a mobile manipulator,” *International Journal of Advanced Robotic Systems*, vol. 9, no. 6, p. 257, 2012. [Online]. Available: https://doi.org/10.5772/53276

[12] A. Dömel, S. Kriegel, M. Kaßecker, M. Brucker, T. Bodenmüller, and M. Suppa, “Toward fully autonomous mobile manipulation for industrial environments,” *International Journal of Advanced Robotic Systems*, vol. 14, no. 4, p. 1729881417718588, 2017. [Online]. Available: https://doi.org/10.1177/1729881417718588

[13] W. Merkt, V. Ivan, Y. Yang, and S. Vijayakumar, “Towards shared autonomy applications using whole-body control formulations of locomanipulation,” in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, Aug 2019, pp. 1206–1211.

[14] A. Dragan, K. Lee, and S. Srinivasa, “Teleoperation with intelligent and customizable interfaces,” *Journal of Human-Robot Interaction, Special Issue on ”Technical and Social Advances in HRI: An Invitational Issue of JHRI”*, vol. 1, no. 3, January 2013.

[15] K.-H. Lee, U. Mehmood, and J.-H. Ryu, “Development of the human interactive autonomy for the shared teleoperation of mobile robots,” *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1524–1529, 2016.

[16] P. Birkenkampf, D. Leidner, and C. Borst, “A knowledge-driven shared autonomy human-robot interface for tablet computers,” *2014 IEEE-RAS International Conference on Humanoid Robots*, pp. 152–159, 2014.

[17] P. Schmaus, D. Leidner, T. Krüger, A. Schiele, B. Pleintinger, R. Bayer, and N. Y. Lii, “Preliminary insights from the meteron supvis justin space-robotics experiment,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3836–3843, Oct 2018.

[18] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*. USA: Wiley-Interscience, 2000.

[19] C. Rösmann, F. Hoffmann, and T. Bertram, “Integrated online trajectory planning and optimization in distinctive topologies,” *Robotics and Autonomous Systems*, vol. 88, pp. 142 – 153, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889016300495

[20] C. Rösmann, F. Hoffmann, and T. Bertram, “Kinodynamic trajectory optimization and control for car-like robots,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 5681–5686.

[21] M. G. Catalano, G. Grioli, E. Farnioli, A. Serio, M. Bonilla, M. Garabini, C. Piazza, M. Gabiccini, and A. Bicchi, *From Soft to Adaptive Synergies: The Pisa/IIT SoftHand*. Cham: Springer International Publishing, 2016, pp. 101–125. [Online]. Available: https://doi.org/10.1007/978-3-319-26706-7_8

[22] NASA, “NASA TLX test,” https://humansystems.arc.nasa.gov/groups/TLX/, 2011, [Online; accessed 25-February-2020].